An Overview of the SPHINX-II Speech Recognition System

Xuedong Huang, Fileno Alleva, Mei-Yuh Hwang, and Ronald Rosenfeld

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

In the past year at Carnegie Mellon steady progress has been made in the area of acoustic and language modeling. The result has been a dramatic reduction in speech recognition errors in the SPHINX-II system. In this paper, we review SPHINX-II and summarize our recent efforts on improved speech recognition. Recently SPHINX-II achieved the lowest error rate in the November 1992 DARPA evaluations. For 5000-word, speaker-independent, continuous, speech recognition, the error rate was reduced to 5%.

1. INTRODUCTION

At Carnegie Mellon, we have made significant progress in large-vocabulary speaker-independent continuous speech recognition during the past years [16, 15, 3, 18, 14]. In comparison with the SPHINX system [23], SPHINX-II offers not only significantly fewer recognition errors but also the capability to handle a much larger vocabulary size. For 5,000-word speaker-independent speech recognition, the recognition error rate has been reduced to 5%. This system achieved the lowest error rate among all of the systems tested in the November 1992 DARPA evaluations, where the testing set has 330 utterances collected from 8 new speakers. Currently we are refining and extending these and related technologies to develop practical unlimited-vocabulary dictation systems, and spoken language systems for general application domains with larger vocabularies and reduced linguistic constraints.

One of the most important contributions to our systems development has been the availability of large amounts of training data. In our current system, we used about 7200 utterances of read Wall Street Journal (WSJ) text, collected from 84 speakers (half male and half female speakers) for acoustic model training; and 45-million words of text published by the WSJ for language model training. In general, more data requires different models so that more detailed acoustic-phonetic phenomena can be well characterized. Towards this end, our recent progress can be broadly classified into feature extraction, detailed representation through parameter sharing, search, and language modeling. Our specific contributions in SPHINX-II include normalized feature representations, multiple-codebook semi-continuous hidden Markov models, between-word senones, multi-pass search algorithms, long-distance language models, and unified acoustic and language representations. The SPHINX-II system block diagram is illustrated in Figure 1, where feature codebooks, dictionary, senones, and language models are iteratively reestimated with the semi-continuous hidden Markov model (SCHMM), albeit not all of them are jointly optimized for the WSJ task at present. In this paper, we will characterize our contributions

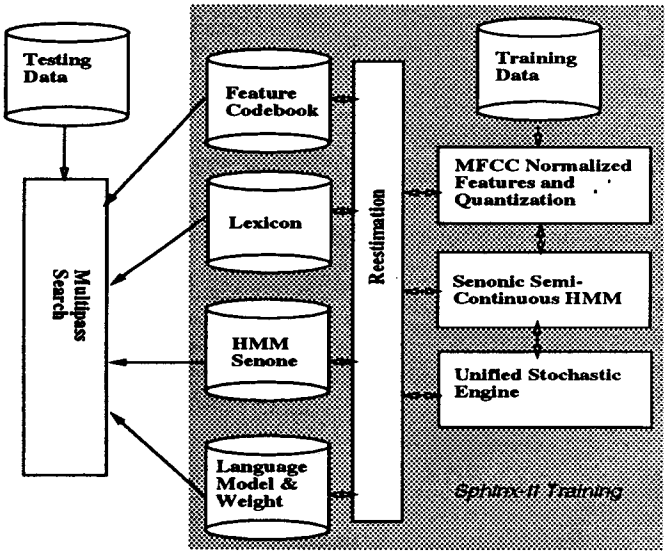

Figure 1: Sphinx-II System Diagram

by percent error rate reduction. Most of these experiments were performed on a development test set for the 5000-word WSJ task. This set consists of 410 utterances from 10 new speakers.

2. FEATURE EXTRACTION

The extraction of reliable features is one of the most important issues in speech recognition and as a result the training data plays a key role in this research. However the curse of dimensionality reminds us that the amount of training data will always be limited. Therefore incorporation of additional features may not lead to any measurable error reduction. This does not necessarily mean that the additional features are poor ones, but rather that we may have insufficient data to reliably model those features. Many systems that incorporate

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1993		**00-00-1993 to 00-00-1993**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
An Overview of the SPHINX-II Speech Recognition System	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University,School of Computer Science,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**6**	

environmentally-robust [1] and speaker-robust [11] models face similar constraints.

2.1. MFCC Dynamic Features

Temporal changes in the spectra are believed to play an important role in human perception. One way to capture this information is to use delta coefficients that measure the change in coefficients over time. Temporal information is particularly suitable for HMMs, since HMMs assume each frame is independent of the past, and these dynamic features broaden the scope of a frame. In the past, the SPHINX system has utilized three codebooks containing [23]: (1) 12 LPC cepstrum coefficients $x_t(k)$, $1 <= k <= 12$; (2) 12 differenced LPC cepstrum coefficients (40 msec. difference) $\Delta x_t(k)$, $1 <= k <= 12$; (3) Power and differenced power (40 msec.) $x_t(0)$ and $\Delta x_t(0)$. Since we are using a multiple-codebook hidden Markov model, it is easy to incorporate new features by using an additional codebook. We experimented with a number of new measures of spectral dynamics, including: (1) second order differential cepstrum and power ($\Delta\Delta x_t(k)$, $1 <= k <= 12$, and $\Delta\Delta x_t(0)$) and third order differential cepstrum and power. The first set of coefficients is incorporated into a new codebook, whose parameters are second order differences of the cepstrum. The second order difference for frame t, $\Delta\Delta x_t(k)$, where t is in units of 10ms, is the difference between $t+1$ and $t-1$ first order differential coefficients, or $\Delta\Delta x_t(k) = \Delta x_{t-1}(k) - \Delta x_{t+1}(k)$. Next, we incorporated both 40 msec. and 80 msec. differences, which represent short-term and long-term spectral dynamics, respectively. The 80 msec. differenced cepstrum $\Delta x'_t(k)$ is computed as: $\Delta x'_t(k) = x_{t-4}(k) - x_{t+4}(k)$. We believe that these two sources of information are more complementary than redundant. We incorporated both Δx_t and $\Delta x'_t$ into one codebook (combining the two into one feature vector), weighted by their variances. We attempted to compute optimal linear combination of cepstral segment, where weights are computed from linear discriminants. But we found that performance deteriorated slightly. This may be due to limited training data or there may be little information beyond second-order differences. Finally, we compared mel-frequency cepstral coefficients (MFCC) with our bilinear transformed LPC cepstral coefficients. Here we observed a significant improvement for the SCHMM model, but nothing for the discrete model. This supported our early findings about problems with modeling assumptions [15]. Thus, the final configuration involves 51 features distributed among four codebooks, each with 256 entries. The codebooks are: (1) 12 mel-scale cepstrum coefficients; (2) 12 40-msec differenced MFCC and 12 80-msec differenced MFCC; (3) 12 second-order differenced MFCC; and (4) power, 40-msec differenced power, second-order differenced power. The new feature set reduced errors by more than 25% over the baseline SPHINX results on the WSJ task.

3. DETAILED MODELING THROUGH PARAMETER SHARING

We need to model a wide range of acoustic-phonetic phenomena, but this requires a large amount of training data. Since the amount of available training data will always be finite one of the central issues becomes that of how to achieve the most detailed modeling possible by means of parameter sharing. Our successful examples include SCHMMs and senones.

3.1. Semi-Continuous HMMs

The semi-continuous hidden Markov model (SCHMM) [12] has provided us with an an excellent tool for achieving detailed modeling through parameter sharing. Intuitively, from the continuous mixture HMM point of view, SCHMMs employ a shared mixture of continuous output probability densities for each individual HMM. Shared mixtures substantially reduce the number of free parameters and computational complexity in comparison with the continuous mixture HMM, while maintaining, reasonably, its modeling power. From the discrete HMM point of view, SCHMMs integrate quantization accuracy into the HMM, and robustly estimate the discrete output probabilities by considering multiple codeword candidates in the VQ procedure. It mutually optimizes the VQ codebook and HMM parameters under a unified probabilistic framework [13], where each VQ codeword is regarded as a continuous probability density function.

For the SCHMM, an appropriate acoustic representation for the diagonal Gaussian density function is crucial to the recognition accuracy [13]. We first performed exploratory semi-continuous experiments on our three-codebook system. The SCHMM was extended to accommodate a multiple feature front-end [13]. All codebook means and covariance matrices were reestimated together with the HMM parameters except the power covariance matrices, which were fixed. When three codebooks were used, the diagonal SCHMM reduced the error rate of the discrete HMM by 10-15% for the RM task [16]. When we used our improved 4-codebook MFCC front-end, the error rate reduction is more than 20% over the discrete HMM.

Another advantage of using the SCHMM is that it requires less training data in comparison with the discrete HMM. Therefore, given the current limitations on the size of the training data set, more detailed models can be employed to improve the recognition accuracy. One way to increase the number of parameters is to use speaker-clustered models. Due to the smoothing abilities of the SCHMM, we were able to train multiple sets of models for different speakers. We investigated automatic speaker clustering as well as explicit male, female, and generic models. By using sex dependent models with the SCHMM, the error rate is further reduced by 10% on the WSJ task.

3.2. Senones

To share parameters among different word models, context-dependent subword models have been used successfully in many state-of-the-art speech recognition systems [26, 21, 17]. The principle of parameter sharing can also be extended to subphonetic models [19, 18]. We treat the state in phonetic hidden Markov models as the basic subphonetic unit — *senone*. Senones are constructed by clustering the state-dependent output distributions across different phonetic models. The total number of senones can be determined by clustering all the triphone HMM states as the shared-distribution models [18]. States of different phonetic models may thus be tied to the same senone if they are close according to the distance measure. Under the senonic modeling framework, we could also use a senonic decision tree to predict unseen triphones. This is particularly important for *vocabulary-independence* [10], as we need to find subword models which are detailed, consistent, trainable and especially generalizable. Recently we have developed a new senonic decision-tree to predict the subword units not covered in the training set [18]. The decision tree classifies senones by asking questions in a hierarchical manner [7]. These questions were first created using speech knowledge from human experts. The tree was automatically constructed by searching for simple as well as composite questions. Finally, the tree was pruned using cross validation. When the algorithm terminated, the leaf nodes of the tree represented the senones to be used. For the WSJ task, our overall senone models gave us 35% error reduction in comparison with the baseline SPHINX results.

The advantages of senones include not only better parameter sharing but also improved pronunciation optimization. Clustering at the granularity of the state rather than the entire model (like generalized triphones [21]) can keep the dissimilar states of two models apart while the other corresponding states are merged, and thus lead to better parameter sharing. In addition, senones give us the freedom to use a larger number of states for each phonetic model to provide more detailed modeling. Although an increase in the number of states will increase the total number of free parameters, with senone sharing redundant states can be clustered while others are uniquely maintained.

Pronunciation Optimization. Here we use the forward-backward algorithm to iteratively optimize a senone sequence appropriate for modeling multiple utterances of a word. To explore the idea, given the multiple examples, we train a word HMM whose number of states is proportional to the average duration. When the Baum-Welch reestimation reaches its optimum, each estimated state is *quantized* with the senone codebook. The closest one is used to label the states of the word HMM. This sequence of senones becomes the senonic baseform of the word. Here arbitrary sequences of senones are allowed to provide the flexibility for the automatically learned pronunciation. When the senone sequence of every word is determined, the parameters (senones) may be re-trained. Although each word model generally has more states than the traditional phoneme-concatenated word model, the number of parameters remains the same since the size of the senone codebook is unchanged. When senones were used for pronunciation optimization in a preliminary experiment, we achieved 10-15% error reduction in a speaker-independent continuous spelling task [19].

4. MULTI-PASS SEARCH

Recent work on search algorithms for continuous speech recognition has focused on the problems related to large vocabularies, long distance language models and detailed acoustic modeling. A variety of approaches based on Viterbi beam search [28, 24] or stack decoding [5] form the basis for most of this work. In comparison with stack decoding, Viterbi beam search is more efficient but less optimal in the sense of MAP. For stack decoding, a fast-match is necessary to reduce a prohibitively large search space. A reliable fast-match should make full use of detailed acoustic and language models to avoid the introduction of possibly unrecoverable errors. Recently, several systems have been proposed that use Viterbi beam search as a fast-match [27, 29], for stack decoding or the N-best paradigm [25]. In these systems, N-best hypotheses are produced with very simple acoustic and language models. A multi-pass rescoring is subsequently applied to these hypotheses to produce the final recognition output. One problem in this paradigm is that decisions made by the initial phase are based on simplified models. This results in errors that the N-best hypothesis list cannot recover. Another problem is that the rescoring procedure could be very expensive per se as many hypotheses may have to be rescored. The challenge here is to design a search that makes the appropriate compromises among memory bandwidth, memory size, and computational power [3].

To meet this challenge we incrementally apply all available acoustic and linguistic information in three search phases. Phase one is a left to right Viterbi Beam search which produces word end times and scores using right context between-word models with a bigram language model. Phase two, guided by the results from phase one, is a right to left Viterbi Beam search which produces word beginning times and scores based on left context between-word models. Phase three is an A* search which combines the results of phases one and two with a long distance language model.

4.1. Modified A* Stack Search

Each theory, *th*, on the stack consists of five entries. A partial theory, *th.pt*, a one word extension *th.w*, a time *th.t* which denotes the boundary between *th.pt* and *th.w*, and two scores *th.g*, which is the score for *th.pt* up to time *th.t* and *th.h* which

is the best score for the remaining portion of the input starting with *th.w* at time *th.t+1* through to the end. Unique theories are determined by *th.pt* and *th.w*. The algorithm proceeds as follows.

1. Add initial states to the stack.
2. According to the evaluation function $th.g+th.h$, remove the best theory, th, from the stack.
3. If th accounts for the entire input then output the sentence corresponding to th. Halt if this is the Nth utterance output.
4. For the word $th.w$ consider all possible end times, t as provided by the left/right lattice.
   (a) For all words, w, beginning at time $t+1$ as provided by the right/left lattice
      i. Extend theory th with w. Designate this theory as *th'*. Set $th'.pt = th.pt + th.w$, $th'.w ::= w$ and $th'.t = t$.
      ii. Compute scores $th'.g = th.g + w\_score(w, th.t + 1, t)$, and $th'.h$. See following for definition of w_score and $th'.h$ computation.
      iii. If th' is already on the stack then choose the best instance of th' otherwise push th' onto the stack.
5. Goto step 2.

4.2. Discussion

When *th* is extended we are considering all possible end times *t* for *th.w* and all possible extensions *w*. When extending *th* with *w* to obtain *th'* we are only interested in the value for *th'.t* which gives the best value for *th'.h + th'.g*. For any *t* and *w*, *th'.h* is easily determined via table lookup from the right/left lattice. Furthermore the value of *th'.g* is given by *th.g + w_score (w, th.t+1, t)*. The function *w_score(w,b,e)* computes the score for the word *w* with begin time *b* and end time *e*.

Our objective is to maximize the recognition accuracy with a minimal increase in computational complexity. With our decomposed, incremental, semi-between-word-triphones search, we observed that early use of detailed acoustic models can significantly reduce the recognition error rate with a negligible increase computational complexity as shown in Figure 2.

By incrementally applying knowledge we have been able to decompose the search so that we can efficiently apply detailed acoustic or linguistic knowledge in each phase. Further more, each phase defers decisions that are better made by a subsequent phase that will apply the appropriate acoustic or linguistic information.

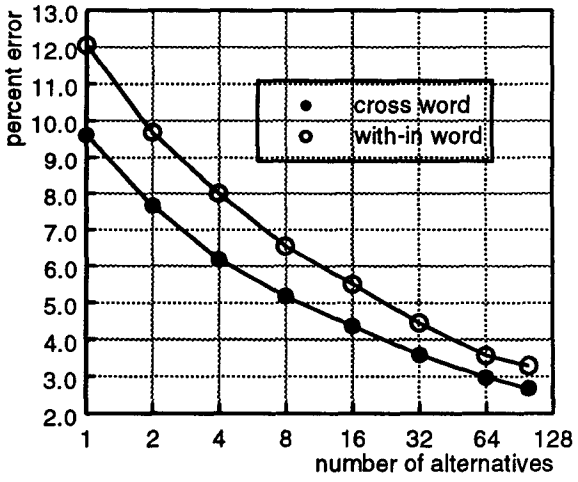

Figure 2: Comparison between early and late use of knowledge.

5. UNIFIED STOCHASTIC ENGINE

Acoustic and language models are usually constructed separately, where language models are derived from a large text corpus without consideration for acoustic data, and acoustic models are constructed from the acoustic data without exploiting the existing text corpus used for language training. We recently have developed a unified stochastic engine (USE) that jointly optimizes both acoustic and language models. As the true probability distribution of both the acoustic and language models can not be accurately estimated, they can not be considered as real probabilities but scores from two different sources. Since they are scores instead of probabilities, the straightforward implementation of the Bayes equation will generally not lead to a satisfactory recognition performance. To integrate language and acoustic probabilities for decoding, we are forced to weight acoustic and language probabilities with a so called language weight [6]. The constant language weight is usually tuned to balance the acoustic probabilities and the language probabilities such that the recognition error rate can be minimized. Most HMM-based speech recognition systems have one single constant language weight that is independent of any specific acoustic or language information, and that is determined using a hill-climbing procedure on development data. It is often necessary to make many runs with different language weights on the development data in order to determine the best value.

In the unified stochastic engine (USE), not only can we iteratively adjust language probabilities to fit our given acoustic representations but also acoustic models. Our multi-pass

search algorithm generates N-best hypotheses which are used to optimize language weights or implement many discriminative training methods, where recognition errors can be used as the objective function [20, 25]. With the progress of new database construction such as DARPA's CSR Phase II, we believe acoustically-driven language modeling will eventually provide us with dramatic performance improvements.

In the N-best hypothesis list, we can assume that the correct hypothesis is always in the list (we can insert the correct answer if it is not there). Let hypothesis be a sequence of words $w_1, w_2, ...w_k$ with corresponding language and acoustic probabilities. We denote the correct word sequence as θ, and all the incorrect sentence hypotheses as $\bar{\theta}$. We can assign a variable weight to each of the n-gram probabilities such that we have a weighted language probability as:

$$W(W) = \prod_i Pr(w_i|w_{i-1}w_{i-2}...)^{\alpha(\mathcal{X}_i, w_i, w_{i-1}, ...)} \quad (1)$$

where the weight $\alpha()$ is a function of acoustic data, $\mathcal{X}_i$, for w_i, and words $w_i, w_{i-1}, ...$ For a given sentence k, a very general objective function can be defined as

$$\begin{aligned} L_k(\lambda) \;=\; & \sum_{\bar{\theta}} Pr(\bar{\theta})\{-\sum_{i\in\theta}[logPr(\mathcal{X}_i|w_i) + \\ & +\alpha(\mathcal{X}_i, w_iw_{i-1}...)logPr(w_i|w_{i-1}w_{i-2}...)] + \\ & +\sum_{i\in\bar{\theta}}[logPr(\mathcal{X}_i|w_i) + \\ & +\alpha(\mathcal{X}_i, w_iw_{i-1}...)logPr(w_i|w_{i-1}...)]\}. \quad (2) \end{aligned}$$

where λ denotes acoustic and language model parameters as well as language weights, $Pr(\bar{\theta})$ denotes the a priori probability of the incorrect path $\bar{\theta}$, and $Pr(\mathcal{X}_i|w_i)$ denotes acoustic probability generated by word model w_i. It is obvious that when $L_k(\lambda) > 0$ we have a sentence classification error. Minimization of Equation 2 will lead to minimization of sentence recognition error rate. To jointly optimize the whole training set, we first define a nondecreasing, differentiable cost function $l_k(\lambda)$ (we use the sigmoid function here) in the same manner as the adaptive probabilistic decent method [4, 20]. There exist many possible gradient decent procedures for the proposed problems.

The term $\alpha(\mathcal{X}_i, w_iw_{i-1}...)logPr(w_i|w_{i-1}...)$ could be merged as one item in Equation 2. Thus we can have language probabilities directly estimated from the acoustic training data. The proposed approach is fundamentally different from traditional stochastic language modeling. Firstly, conventional language modeling uses a text corpus only. Any acoustical confusable words will not be reflected in language probabilities. Secondly, maximum likelihood estimation is usually used, which is only loosely related to minimum sentence error. The reason for us to keep $\alpha()$ separate from the language probability is that we may not have sufficient acoustic data to estimate the language parameters at present. Thus, we are forced to have $\alpha()$ shared across different words so we may have n-gram-dependent, word-dependent or even word-count-dependent language weights. We can use the gradient decent method to optimize all of the parameters in the USE. When we jointly optimize $L(\lambda)$, we not only obtain our unified acoustic models but also the unified language models. A preliminary experiment reduced error rate by 5% on the WSJ task [14]. We will extend the USE paradigm for joint acoustic and language model optimization. We believe that the USE can further reduce the error rate with an increased amount of training data.

6. LANGUAGE MODELING

Language Modeling is used in Sphinx-II at two different points. First, it is used to guide the beam search. For that purpose we used a conventional backoff bigram for that purpose. Secondly, it is used to recalculate linguistic scores for the top N hypotheses, as part of the N-best paradigm. We concentrated most of our language modeling effort on the latter.

Several variants of the conventional backoff trigram language model were applied at the reordering stage of the N-best paradigm. (Eventually we plan to incorporate this language model into the A* phase of the multi-pass search with the USE). The best result, a 22% word error rate reduction, was achieved with the simple, non-interpolated "backward" trigram, with the conventional forward trigram finishing a close second.

7. SUMMARY

Our contributions in SPHINX-II include improved feature representations, multiple-codebook semi-continuous hidden Markov models, between-word senones, multi-pass search algorithms, and unified acoustic and language modeling. The key to our success is our data-driven unified optimization approach. This paper characterized our contributions by percent error rate reduction on the 5000-word WSJ task, for which we reduced the word error rate from 20% to 5% in the past year [2].

Although we have made dramatic progress there remains a large gap between commercial applications and laboratory systems. One problem is the large number of out of vocabulary (OOV) words in real dictation applications. Even for a 20000-word dictation system, on average more than 25% of the utterances in a test set contain OOV words. Even if we exclude those utterance containing OOV words, the error rate is still more than 9% for the 20000-word task due to the limitations of current technology. Other problems are illustrated by the November 1992 DARPA stress test evaluation, where testing data comprises both spontaneous speech with many OOV words but also speech recorded using several different microphones. Even though we augmented our system with

more than 20,000 utterances in the training set and a noise normalization component [1], our augmented system only reduced the error rate of our 20000-word baseline result from 12.8% to 12.4%, and the error rate for the stress test was even worse when compared with the baseline (18.0% vs. 12.4%). To summarize, our current word error rates under different testing conditions are listed in Table 1. We can see from this

Systems	Vocabulary	Test Set	Error Rate
Baseline	5000	330 utt.	5.3%
Baseline	20000	333 utt.	12.4%
Stress Test	20000	320 utt.	18.0%

Table 1: Performance of SPHINX-II in real applications.

table that improved modeling technology is still needed to make speech recognition a reality.

8. Acknowledgements

This research was sponsored by the Defense Advanced Research Projects Agency and monitored by the Space and Naval Warfare Systems Command under Contract N00039-91-C-0158, ARPA Order No. 7239.

The authors would like to express their gratitude to Raj Reddy and other members of CMU speech group for their help.

References

1. Acero, A. *Acoustical and Environmental Robustness in Automatic Speech Recognition.* Department of Electrical Engineering, Carnegie-Mellon University, September 1990.
2. Alleva, F., Hon, H., Huang, X., Hwang, M., Rosenfeld, R., and Weide, R. *Applying SPHINX-II to the DARPA Wall Street Journal CSR Task.* in: **DARPA Speech and Language Workshop**. Morgan Kaufmann Publishers, San Mateo, CA, 1992.
3. Alleva, F., Huang, X., and Hwang, M. *An Improved Search Algorithm for Continuous Speech Recognition.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1993.
4. Amari, S. *A Theory of Adaptive Pattern Classifiers.* **IEEE Trans. Electron. Comput.**, vol. EC-16 (1967), pp. 299–307.
5. Bahl, L. R., Jelinek, F., and Mercer, R. *A Maximum Likelihood Approach to Continuous Speech Recognition.* **IEEE Transactions on Pattern Analysis and Machine Intelligence**, vol. PAMI-5 (1983), pp. 179–190.
6. Bahl, L., Bakis, R., Jelinek, F., and Mercer, R. *Language-Model/Acoustic Channel Balance Mechanism.* **IBM Technical Disclosure Bulletin**, vol. 23 (1980), pp. 3464–3465.
7. Breiman, L., Friedman, J., Olshen, R., and Stone, C. **Classification and Regression Trees**. Wadsworth, Inc., Belmont, CA., 1984.
8. Hon, H. and Lee, K. *CMU Robust Vocabulary-Independent Speech Recognition System.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. Toronto, Ontario, CANADA, 1991, pp. 889–892.
9. Huang, X. *Minimizing Speaker Variations Effects for Speaker-Independent Speech Recognition.* in: **DARPA Speech and Language Workshop**. Morgan Kaufmann Publishers, San Mateo, CA, 1992.
10. Huang, X. *Phoneme Classification Using Semicontinuous Hidden Markov Models.* **IEEE Transactions on Signal Processing**, vol. 40 (1992), pp. 1062–1067.
11. Huang, X., Ariki, Y., and Jack, M. **Hidden Markov Models for Speech Recognition**. Edinburgh University Press, Edinburgh, U.K., 1990.
12. Huang, X., Belin, M., Alleva, F., and Hwang, M. *Unified Stochastic Engine (USE) for Speech Recognition.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1993.
13. Huang, X., Hon, H., Hwang, M., and Lee, K. *A Comparative Study of Discrete, Semicontinuous, and Continuous Hidden Markov Models.* **Computer Speech and Language, in press**, 1993.
14. Huang, X., Lee, K., Hon, H., and Hwang, M. *Improved Acoustic Modeling for the SPHINX Speech Recognition System.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. Toronto, Ontario, CANADA, 1991, pp. 345–348.
15. Hwang, M., Hon, H., and Lee, K. *Modeling Between-Word Coarticulation in Continuous Speech Recognition.* in: **Proceedings of Eurospeech**. Paris, FRANCE, 1989, pp. 5–8.
16. Hwang, M. and Huang, X. *Shared-Distribution Hidden Markov Models for Speech Recognition.* **IEEE Transactions on Speech and Audio Processing**, vol. 1 (1993).
17. Hwang, M. and Huang, X. *Subphonetic Modeling with Markov States — Senone.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1992.
18. Juang, B.-H. and Katagiri, S. *Discriminative Learning for Minimum Error Classification.* **IEEE Trans on Signal Processing, to appear**, December 1992.
19. Lee, K. *Context-Dependent Phonetic Hidden Markov Models for Continuous Speech Recognition.* **IEEE Transactions on Acoustics, Speech, and Signal Processing**, April 1990, pp. 599–609.
20. Lee, K., Hon, H., and Reddy, R. *An Overview of the SPHINX Speech Recognition System.* **IEEE Transactions on Acoustics, Speech, and Signal Processing**, January 1990, pp. 35–45.
21. Lowerre, B. and Reddy, D. *The Harpy Speech Understanding System.* in: **The Harpy Speech Understanding System**, by B. Lowerre and D. Reddy, edited by W. Lee. Prentice-Hall, Englewood Cliffs, NJ, 1980.
22. Schwartz, R., Austin, S., Kubala, F., and Makhoul, J. *New Uses for the N-Best Sentence Hypotheses Within the Byblos Speech Recognition System.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1992, pp. 1–4.
23. Schwartz, R., Chow, Y., Kimball, O., Roucos, S., Krasner, M., and Makhoul, J. *Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1985, pp. 1205–1208.
24. Soong, F. and Huang, E. *A Tree-Trellis Based Fast Search for Finding the N-Best Sentence Hypothesis.* in: **DARPA Speech and Language Workshop**. 1990.
25. Viterbi, A. J. *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm.* **IEEE Transactions on Information Theory**, vol. IT-13 (1967), pp. 260–269.